(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（A）**　(11)特許出願公開番号

特開平8－44563

(43)公開日　平成８年(1996)２月16日

(51)Int.Cl.[6]		識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｇ０６Ｆ	9/38	３７０　Ｂ			
		３１０　Ａ			
	12/08		Ｈ　7623－5Ｂ		

審査請求　未請求　請求項の数１　ＯＬ　（全 11 頁）

(21)出願番号　特願平6－178129

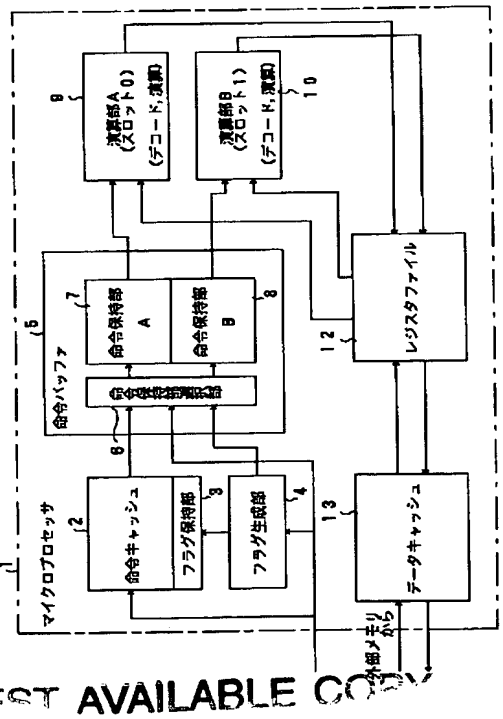
(22)出願日　平成６年(1994)７月29日

(71)出願人　000005223
富士通株式会社
神奈川県川崎市中原区上小田中1015番地
(72)発明者　久保沢　元
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内
(74)代理人　弁理士　長谷川　文廣　（外２名）

(54)【発明の名称】　マイクロプロセッサ

(57)【要約】
【目的】　１クロックサイクルに複数の命令を同時に実行できるマイクロプロセッサに関し，デコードステージでのスロット選択を行わないようにして，高速化を図ることを目的とする。
【構成】　演算処理する命令を保持する命令バッファと命令バッファから命令を入力して演算処理する複数の演算部とを備え，複数の命令実行を同時にできるマイクロプロセッサにおいて，命令は処理する演算部が定められたものであり，命令バッファは，それぞれの演算部に対応して命令保持部を備え，命令が使用する演算部を特定するフラグを生成して命令に付加するフラグ生成部と，フラグに応じて命令バッファの該命令保持部を選択する命令保持選択部とを備え，命令が処理する演算部に対応した命令保持部に該命令を保持する構成を持つ。

本発明の基本構成



【特許請求の範囲】
【請求項１】　演算処理する命令を保持する命令バッファと命令バッファから命令を入力して演算処理する複数の演算部とを備え，複数の命令実行を同時にできるマイクロプロセッサにおいて，命令は処理される演算部が定められたものであり，命令バッファは，それぞれの演算部に対応して命令保持部を備え，命令が使用する演算部を特定するフラグを生成して命令に付加するフラグ生成部と，該フラグに応じて命令バッファの該命令保持部を選択する命令保持部選択部とを備え，命令が処理される演算部に対応した命令保持部に該命令を保持するようにしたことを特徴とするマイクロプロセッサ。
【発明の詳細な説明】
【０００１】
【産業上の利用分野】本発明は，マイクロプロセッサに関する。特に，複数の演算ユニットを有し，１クロックサイクルに複数の命令を同時に実行できるマイクロプロセッサに関する。
【０００２】パイプラインで複数の命令を同時に実行することにより，命令を一つずつ順次実行する場合と比較して高速にプログラム処理することができる。パイプライン処理により複数命令を同時に実行するためには，命令をどの演算器で実行すれば良いかを判定して適切なパイプライン（スロット）に命令を転送する必要がある。
【０００３】本発明は，パイプラインのどのスロットに命令を発行するかの判定とスロットへの命令転送を効率良く行うことのできるマイクロプロセッサを提供する。
【０００４】
【従来の技術】図８は従来のパイプライン処理をするマイクロプロセッサの構成を示す。２００はプロセッサである。
【０００５】２１０は入力バッファであって，外部メモリから入力される命令を一時保持するものである。２１１は命令キャッシュであって，命令を保持するキャッシュである。
【０００６】２１２はデータキャッシュであって，データを保持するキャッシュである。２１３は命令バッファであって，命令をパイプライン処理のスロット（スロット０，スロット１）に入力するためのバッファである。
【０００７】２１４はスロット０であって，パイプライン処理するものである。２１５はスロット１であって，パイプライン処理するものである。２１６はレジスタファイルであって，演算器の入力データ（オペランド），演算器の演算結果等を一時的に保持する複数のレジスタにより構成されるものである。
【０００８】２２０，２２１はデコーダであって，命令をデコードするものである。２３０は演算器１であって，スロット０の演算処理をするものである。２３１は演算器２であって，スロット０の演算処理をするものである。
【０００９】２３２は演算器３であって，スロット１の演算処理をするものである。２３３は演算器４であって，スロット１の演算処理をするものである。図８の従来のマイクロプロセッサの構成の動作を説明する。
【００１０】命令は，命令バッファ２１３から１サイクルに２命令ずつデコーダ２２０，２２１に供給される。命令バッファに命令がある場合は，そこから命令が読み出され，スロット０，スロット１の各デコーダ２２０，２２１で解釈される。１命令の語長を４バイトとすると，命令バッファへ２１３の書き込み，読み出しは８バイト単位で行われる。命令キャッシュ２１１への書き込み，読み出しの単位も８バイトとする。
【００１１】命令バッファ２１３に命令がないとき，命令が命令キャッシュ２１１から読み出されて命令バッファ２１３に書き込まれる（命令フェッチ）。そして，命令バッファ２１３の命令はスロット０もしくはスロット１に入力され，デコーダ２２０，２２１で解釈される。デコードステージでは命令間のデータ依存性，演算器等の資源の競合がチェックされ，競合がある場合にはそれが解消するまで演算の開始を待たせる。デコードされた命令は，各命令に応じた演算器（演算器１，演算器２，演算器３，演算器４）で演算される。演算を行うための入力オペランドはデコードと並行してレジスタファイル２１６から読み出される。
【００１２】命令が命令キャッシュ２１１に存在しない場合には，マイクロプロセッサ２００の外部メモリから読み出された命令が命令バッファ２１３に書き込まれるとともに，命令キャッシュ２１１にも書き込まれる。命令が演算ステージ（演算器１，演算器２，演算器３，演算器４）に移行した後のフローは命令によって異なる。演算が算術演算，論理演算の場合は，１サイクルで演算は終了し，結果は３サイクル後にレジスタファイル２１６に書き込まれる。命令がロード命令の場合は，演算実行ステージでメモリにアクセスするアドレスを計算し，次のキャッシュステージでデータキャッシュを読み出す。次のステージで例外処理（例えば，アクセスを禁止されているアドレスが求められてはいないかを検出する等の処理）を行い，次のステージでレジスタの内容を更新する。
【００１３】図８において，演算器１，演算器２，演算器３，演算器４はすべて異なる機能の演算を行うものとする。つまり，同じ種類の演算を２命令同時に実行することはできない。また，スロット０にある命令はスロット１側の演算器を使用することはできない。同様に，スロット１にある命令はスロット０で使用することはできない。これらはスロット間の配線およびマルチプレクサが増加し制御が複雑になることを避けるためである。
【００１４】通常は，命令キャッシュ２１１からバッファ２１３に命令を転送してきた時点では，スロット０にはフェッチしてきた８バイトの下位側の命令が，スロッ

ト１には上位の命令が入っている。各演算器（２３０，２３１，２３２，２３３）に上記のような制限があった場合には，命令がどの演算器を使用するのかを検出して適切なスロットが選ばれる必要がある。使用する演算器と違うスロットに入っている場合には，命令スロットの入れ替えが必要になる。この入れ替えは命令バッファから命令を読み出し，デコードして演算器に渡されるまでの間（デコードステージ）に行わなければならない。

【００１５】

【発明が解決しようとする課題】従来のパイプライン処理するマイクロプロセッサはデコードステージで命令スロットの入れ替えを行うようにしていた。デコードステージは命令のデコード，依存関係のチェック，レジスタファイル読み出し，命令発行が可能であるか等の判定を行うのでプロセッサ全体のクリティカルパス（動作遅延の最も大きいパス）になる可能性の高いステージである。デコードステージで命令がどの演算器を使用するかを判定し，適切なスロットを選択することはマイクロプロセッサの高速動作をさまたげる大きな要因となる。

【００１６】本発明は，デコードステージでのスロット選択を行わないようにして，マイクロプロセッサの高速化を図ることを目的とする。

【００１７】

【課題を解決するための手段】本発明は，命令キャッシュにチップ外部から命令を格納する際に，命令がどのスロットを使用するかを示すフラグを付け，命令キャッシュにそのフラグとともに書き込む。また，外部メモリから読み出した命令を命令バッファに書き込む場合も同様に，フラグを付け，命令バッファと命令キャッシュに書き込むようにする。そして，命令バッファへの書き込みの際に命令キャッシュのフラグを見て適切なスロットを選択し，命令バッファ内の転送制御はこのスロットフラグによって行うことにより，デコードステージでスロット選択を行わないでもスロットの選択をできるようにした。

【００１８】図１は本発明の基本構成を示す。図１は二つのスロットを備えている場合を例として示す。図１において，１はマイクロプロセッサである。

【００１９】２は命令キャッシュである。３は命令キャッシュのフラグ保持部であって，命令を演算する演算部（スロット）を識別するためのフラグを保持するものである。

【００２０】４はフラグ生成部であって，命令の種類に応じてフラグを生成するものである。５は命令バッファであって，命令キャッシュもしくは外部メモリから読み出した命令を保持するものである。

【００２１】６は命令保持部選択部であって，命令のフラグを識別して，命令保持部Ａもしくは命令保持部Ｂのいずれに命令を保持すべきかを判定するものである。７は命令保持部Ａであって，演算部Ａ（スロット０）で処理される命令を保持するものである。

【００２２】８は命令保持部Ｂであって，演算部Ｂ（スロット１）の側で処理される命令を保持するものである。９は演算部Ａ（スロット０）であって，命令保持部Ａから入力される命令をデコードし，デコードの結果によりレジスタファイル１２からオペランドを入力し，演算し，演算結果をレジスタスタファイル１２に転送するものである。

【００２３】１０は演算部Ｂ（スロット１）であって，命令保持部Ｂから入力される命令をデコードし，デコードした結果によりレジスタファイル１２からオペランドを入力し，演算し，演算結果をレジスタファイル１２に転送するものである。

【００２４】１２はレジスタファイルであって，データキャッシュ１３から入力されるデータもしくは演算部Ａ，演算部Ｂの演算結果を保持するものである。１３はデータキャッシュであって，データを保持するものである。

【００２５】

【作用】図１の本発明の基本構成の動作を説明する。フラグ生成部４は外部メモリから入力される命令をデコードし，命令の種類を判定する。そして，その命令が演算部Ａで処理すべきものであるか，あるいは演算部Ｂで処理すべきものであるかを識別するためのフラグを生成する。例えば，演算部Ａで処理する命令に対しては「０」，演算部Ｂで処理する命令に対しては「１」等である。

【００２６】命令バッファ５に命令キャッシュ２から命令が読み出されて保持される。もしくは，命令キャッシュ２に命令が無い場合に外部メモリから入力される命令はフラグ生成部４でフラグを付され命令バッファ５に入力されるとともに命令キャッシュ２にも保持される。その際，命令保持部選択部６はフラグを識別し，例えば，フラグが「０」であれば命令バッファ５を選択し，命令バッファＡに命令を保持する。あるいは，フラグが「１」であれば，命令保持部選択部６は命令保持部Ｂを選択し，命令保持部Ｂに命令を保持する。レジスタファイル１２はデータキャッシュ１３からオペランドとなるデータを取り出して保持する。

【００２７】演算部Ａには命令保持部Ａの命令が入力され，入力された命令をデコードする。そして，演算部Ａはレジスタファイル１２から必要なオペランドを取り出し演算し，演算結果をレジスタファイル１２に保持させる。同様に，演算部Ｂには命令保持部Ｂの命令が入力され，入力された命令をデコードする。そして，演算部Ｂはレジスタファイル１２から必要とするオペランドを取り出して演算する。そして，演算結果をレジスタファイル１２に保持する。レジスタファイル１２に保持された演算結果はデータキャッシュ１３に転送され，外部メモリに出力される。

【００２８】本発明によれば，マイクロプロセッサの処理速度に大きく影響する演算部Ａ，演算部Ｂのデコードステージでのスロット選択を行わない。そのため，デコードステージで処理が遅延されることがないので，能率的にパイプライン処理を行うことができ，マイクロプロセッサ全体の動作が高速化される。また，命令バッファの命令処理もフラグにより簡単に制御できる。

【００２９】

【実施例】図２は本発明の実施例を示す。図２において，５１はマイクロプロセッサである。

【００３０】５２は命令キャッシュである。５３はスロットを識別するためのフラグである。フラグ「０」でスロット０，フラグ「１」でスロット１を識別するとする。

【００３１】５４はプリデコーダであって，外部メモリから命令をデコードしてスロット０もしくはスロット１のいずれで演算すべき命令であるかを識別し，フラグを生成するものである（図１のフラグ生成部に相当する）。

【００３２】５５は命令バッファである（命令バッファ５５の詳細は図４で説明する）。５６は命令保持部選択部である。５７は命令保持部である。

【００３３】５９はスロット０であって，デコーダと演算器１，演算器２により構成されるものである。６０はスロット１であって，デコーダと演算器３，演算器４により構成されるものである。

【００３４】６２はレジスタファイルである。６３はデータキャッシュである。７６は入力バッファ（外部入力レジスタ）であって，外部メモリから入力される命令を一時保持するバッファである。

【００３５】スロット０において，７０はデコーダであって，スロット０に入力される命令をデコードするものである。

【００３６】７２は演算器１であって，デコーダ７０のデコードした命令の内容に従って，レジスタファイル６２からオペランドを取り出し演算処理するものである。７３は演算器２であって，デコーダ７０のデコードした命令の内容に従って，レジスタファイル６２からオペランドを取り出し演算処理するものである。

【００３７】スロット１において，７１はデコーダであって，スロット１に入力される命令をデコードするものである。

【００３８】７４は演算器３であって，デコーダ７１のデコードした命令の内容に従って，レジスタファイル６２からオペランドを取り出し演算処理するものである。７５は演算器４であって，デコーダ７１のデコードした命令の内容に従って，レジスタファイル６２からオペランドを取り出し演算処理するものである。

【００３９】８０はフラグを付された命令のデータ構造の例である。命令０はスロット０で処理される命令であり，スロット０を識別するフラグを持つ。命令１はスロット１で処理される命令であり，スロット１を識別するフラグを持つ。

【００４０】命令キャッシュ５２の１ラインが３２バイト長であるとすると８命令分が格納できる。フラグの書き込みは，外部メモリから命令を取ってきた時に行う。外部メモリから読み込んだ命令は，命令キャッシュ５２に書き込む前に入力バッファ７６（外部入力レジスタ）に一旦書き込みを行う。フラグの発生はこの入力バッファ７６を読み出してから命令キャッシュ５２に書き込む間にデコーダ５４で行われる。命令キャッシュ５２に書き込むデータが必要になるタイミングはサイクルの後半である。フラグ発生はこれより速く終わるのでプリデコーダ５４によるデコードが動作遅延の原因にはならない（この点については図３で詳述する）。命令キャッシュ５２から命令バッファ５５への読み出しは命令フェッチステージで行う。命令フェッチスタージでは命令キャッシュ５２の読み出しが行われ，命令バッファ５５に書き込まれる。命令バッファ５５において，命令フラグを識別し，命令がスロット０で処理されるべきものであるか，あるいはスロット１で処理されるべきものであるかを命令保持部選択部５６で識別し，スロット０で演算すべき命令であれば命令保持部５７のスロット０側に保持し，あるいはスロット１で演算すべき命令であればスロット１側に保持する。

【００４１】図３は本発明のフラグ生成のタイムチャートである。外部メモリから読み出された命令は一旦，外部入力レジスタ（入力バッファ７６）に保持される。

【００４２】第１サイクルにおいて，まず，第１サイクルのクロックの立ち上がりで，入力レジスタ７６から命令が読み出される。そして，チップ内（マイクロプロセッサ５１の内部）を転送され，プリデコーダ５４でプリデコードされ，第１サイクルの後半で，チップ内を転送されて命令キャッシュ５２の入口の命令キャッシュ入力レジスタ（命令キャッシュ５２に命令を入力するために一時命令を保持するレジスタ（図示せず））に書き込まれる。ここで，命令バッファ５５にも書き込まれる。

【００４３】第２サイクルにおいて，第２サイクルのクロックの立ち上がりで，命令キャッシュの入力レジスタから命令が読み出され、命令キャッシュ５２に書き込まれる。

【００４４】図４は本発明の命令バッファの実施例である。図４において，５２は命令キャッシュである。命令キャッシュには４バイトの命令を８バイト単位に保持される。８バイト構成のＬＳＷ（下位）側の命令（命令０）とＭＳＷ（上位）側の命令（命令１）にそれぞれスロットを識別するフラグが備えられる。

【００４５】５５は命令バッファであり，命令の保持部を７個（命令保持部Ｔ，Ｐ０，Ｐ１，Ｆ０，Ｆ１，Ｄ０，Ｄ１）備えるものである。命令は，命令の連続性，

フラグを考慮した選択ルールに従って選択的に命令保持部に保持される。選択ルールについては後述する。

【００４６】８５は命令保持部選択部１であって，命令キャッシュ５２もしくは命令保持部Ｔから取り出された命令のフラグを参照し，選択ルールに従って命令保持部Ｐ０もしくはＰ１を選択するものである。

【００４７】８６は命令保持部選択部２であって，命令キャッシュ５２，命令保持部Ｐ０もしくは命令保持部Ｐ１から取り出された命令のフラグを参照し，選択ルールに従って命令保持部Ｆ０もしくはＦ１を選択するものである。

【００４８】８７は命令保持部選択部３であって，命令キャッシュ５２，命令保持部Ｆ０もしくは命令保持部Ｆ１から取り出された命令のフラグを参照し，選択ルールに従って命令保持部Ｄ０もしくはＤ１を選択するものである。

【００４９】９０はセレクタ０であって，Ｐ０に保持する命令を選択するものである。９１はセレクタ１であって，Ｐ１に保持する命令を選択するものである。９２はセレクタ２であって，Ｆ０に保持する命令を選択するものである。

【００５０】９３はセレクタ３であって，Ｆ１に保持する命令を選択するものである。９４はセレクタ４であって，Ｄ０に保持する命令を選択するものである。９５はセレクタ５であって，Ｄ１に保持する命令を選択するものである。

【００５１】Ｔは命令保持部であって，命令キャッシュ５２の命令の上位４バイトの命令を保持するものである。Ｐ０は命令保持部であって，セレクタ０（９０）で選択された命令を保持するものである。

【００５２】Ｐ１は命令保持部であって，セレクタ１（９１）で選択された命令を保持するものである。Ｆ０は命令保持部であって，セレクタ２（９２）で選択された命令を保持するものである。

【００５３】Ｆ１は命令保持部であって，セレクタ３（９３）で選択された命令を保持するものである。Ｄ０は命令保持部であって，セレクタ４（９４）で選択された命令を保持するものである。

【００５４】Ｄ１は命令保持部であって，セレクタ５（９５）で選択された命令を保持するものである。図４の構成において，セレクタ０（９０）には，命令保持部Ｔ，命令キャッシュのＬＳＷ（命令０），ＭＳＷ（命令１）の内容とフラグが入力され，フラグおよび選択ルールに従って，それらの命令のいずれかを選択する。そして，Ｐ０に選択した命令０を保持する。同様に，セレクタ１（９１）には，命令保持部Ｔ，命令キャッシュ５２の命令のＬＳＷ，ＭＳＷの内容とフラグが入力され，フラグおよび選択ルールに従って選択されてＰ１に保持される。

【００５５】また，セレクタ２（９２）には，命令保持部Ｐ０，命令キャッシュ５２のＬＳＷ，ＭＳＷの内容とそのフラグが入力され，フラグと選択ルールに従って，そのいずれかを選択し，命令保持部Ｆ０に保持する。同様に，セレクタ３（９３）には，命令保持部Ｐ１，命令キャッシュ５２のＬＳＷ，ＭＳＷの内容とフラグが入力され，フラグと選択ルールに従って命令が選択されて命令保持部Ｆ１に保持される。

【００５６】また，セレクタ４（９４）には，命令保持部Ｆ０，命令キャッシュ５２のＬＳＷ，ＭＳＷの内容とフラグが入力され，フラグと選択ルールに従って命令が選択されて，命令保持部Ｄ０に保持される。同様に，セレクタ５（９５）には，命令保持部Ｆ１，命令キャッシュ５２のＬＳＷ，ＭＳＷの内容とフラグが入力され，フラグと選択ルールに従って命令が選択されて命令保持部Ｄ１に保持される。

【００５７】Ｄ０の内容はスロット０に転送され，Ｄ１の内容はスロット１に転送される。次に図４の構成における命令の転送ルールについて説明する。図４の構成において，各選択部は次のルールに従って，命令を選択し，選択した保持部に転送する。

【００５８】(1) キャッシュからの命令の読み出しは２命令単位で行う。

(2) Ｄ０，Ｆ０，Ｐ０にはスロット０の演算器のグループで実行される命令が入る。

【００５９】(3) Ｄ１，Ｆ１，Ｐ１にはスロット１の演算器のグループで実行される命令が入る。

(4)　読み出した命令がスロット０で実行する命令とスロット１で実行する命令のペアの場合において，(4) －１　命令バッファが空の時はＤ０，Ｄ１に書き込む。

【００６０】(4) －２　Ｄ０，Ｄ１に命令が存在している時はＦ０，Ｆ１に書き込む。

(4) －３　Ｄ０，Ｄ１，Ｆ０，Ｆ１に命令が存在している時はＰ０，Ｔに書き込む。

【００６１】(5)　読み出した命令が両方ともスロット０で実行する命令の場合，(5) －１　命令バッファが空の時はＤ０，Ｆ０に書き込む。(5) －２　Ｄ０，Ｄ１に命令が存在している時はＦ０，Ｐ０に書き込む。

【００６２】(5) －３　Ｄ０，Ｄ１，Ｆ０，Ｆ１に命令が存在している時はＰ０，Ｔに書き込む。

(6)　読み出した命令が両方ともスロット１で実行する命令の場合，(6) －１　命令バッファが空の時はＤ１，Ｆ１に書き込む。

【００６３】(6) －２　Ｄ０，Ｄ１に命令が存在している時はＦ１，Ｐ１に書き込む。(6) －３　Ｄ０，Ｄ１，Ｆ０，Ｆ１に命令が存在している時はＰ１，Ｔに書き込む。

【００６４】(7)　Ｄ０，Ｄ１から命令が発行されると，Ｆ０，Ｆ１，Ｐ０，Ｐ１，Ｔに存在する命令はＤ０，Ｄ１の方向に詰めるようにバッファ内を移動する。

(8)　Ｄ０，Ｄ１に同時に入る命令は連続したアドレス

の命令でなければならない。
【００６５】(9)　Ｆ０，Ｆ１に同時に入る命令は連続したアドレスの命令でなければならない。
(10)　Ｐ０，Ｐ１に同時に入る命令は連続したアドレスの命令でなければならない。
【００６６】図４の命令バッファの動作を動作例に基づいて説明する（図５，図６，図７を参照する）。図５は命令キャッシュ５２に保持されている命令列の例である。４バイト構成の２命令を１サイクルの処理単位とし，８バイトの上位４バイトのＭＳＷと下位４バイトのＬＳＷをそれぞれ１命令とする。それぞれにスロットを識別するフラグを備えている。フラグ「０」はスロット０で処理される命令であり，フラグ「１」はスロット「１」で処理される命令である（ａｄｄ命令はスロット０で実行され，ｌｏａｄ命令はスロット１で実行されたものとする。）
命令キャッシュ５２から各クロックサイクル（以下サイクルと略称する）毎に２命令もしくは，上位バイト（ＭＳＷ）が読み出される。同じサイクルでバッファ（命令保持部）に書き込まれた命令は，同時に実行できる。バッファに書き込まれるサイクルが異なる場合は，命令のアドレスが４バイト境界で連続している命令は同時に実行できる（例えば，上記の命令で，ｌｏａｄ２とａｄｄ３，ａｄｄ４とｌｏａｄ３等）。そうでないもの同士（例えば，ｌｏａｄ２とｌｏａｄ４等）は同時に実行できない。
【００６７】図６は動作例１である。サイクル１で命令キャッシュ５２からａｄｄ１とｌｏａｄ１が読み出され，セレクタ４（９４）とセレクタ５（９５）で選択されてＤ０とＤ１に書き込まれる。ここで，２命令を実行する命令が発行される（２命令発行）。
【００６８】サイクル２で，２命令発行に従って，ａｄｄ１とｌｏａｄ１がそれぞれＤ０，Ｄ１からスロット０，スロット１に出力される。そして，新たにｌｏａｄ２とａｄｄ２が命令キャッシュ５２から読み出され，セレクタ４（９４），セレクタ５（９５）で選択され，それぞれＤ０，Ｄ１に書き込まれる。ここで２命令実行が発行される。
【００６９】サイクル３で，２命令発行に従って，ｌｏａｄ２とａｄｄ２がそれぞれスロット０，スロット１に出力される。そして，新たにａｄｄ３とａｄｄ４が命令キャッシュ５２から読み出される。両方ともスロット０の命令であるので，セレクタ４（９４），セレクタ２（９２）で選択されてＤ０，Ｆ０に書き込まれる。ここで１命令の実行が発行される（１命令発行）。
【００７０】サイクル４で，１命令発行に従ってａｄｄ３がＤ０からスロット０に出力され，セレクタ４（９４）で選択されてａｄｄ４がＦ０からＤ０に移動する。ここで，次の命令ｌｏａｄ３とｌｏａｄ４が命令キャッシュ５２から読み出される。ｌｏａｄ３とａｄｄ４は境界が連続している（プログラムカウンタ値が連続している）ので，同時に実行できる。そのため，ｌｏａｄ３はセレクタ５（９５）で選択されてＤ１に保持され，ｌｏａｄ４はセレクタ３（９３）で選択されてＦ１に保持される。ここで，２命令実行が発行される。
【００７１】サイクル５で，２命令発行のためｌｏａｄ３とａｄｄ４がそれぞれスロット０，スロット１に出力される。そして，セレクタ５（９５）で選択されてｌｏａｄ４がＦ１からＤ１に移動し，新たに命令キャッシュ５２からｌｏａｄ５，ａｄｄ５が読み込まれる。そして，Ｄ１のｌｏａｄ４とａｄｄ５は同時に実行できないので，ａｄｄ５はセレクタ２（９２）で選択されてＦ０に書き込まれ，ｌｏａｄ５はセレクタ３（９３）で選択されてＦ１に書き込まれる。ここで１命令実行が発行される。
【００７２】サイクル６で，１命令発行のためｌｏａｄ４がＤ１からスロット１に出力され，ｌｏａｄ５，ａｄｄ５がそれぞれセレクタ５（９５），セレクタ４（９４）で選択されてＤ１，Ｄ０に移動する。新たに，命令キャッシュ５２から，ｌｏａｄ６，ａｄｄ６が読み出され，それぞれセレクタ３（９３），セレクタ２（９２）で選択されてＦ１とＦ０に書き込まれる。ここで，２命令実行が発行され，次のサイクル（図示せず）でｌｏａｄ５とａｄｄ５が出力される。
【００７３】図７は動作例２を示す。以下，セレクタの選択処理についての説明は省略する。サイクル１で命令キャッシュからａｄｄ１とｌｏａｄ１が読み出され，Ｄ０，Ｄ１に書き込まれる。ここで，０命令発行で，次のサイクルで２命令とも実行されないとする。
【００７４】サイクル２でａｄｄ１とｌｏａｄ１はＤ０とＤ１に保持される。そして，次の命令（ｌｏａｄ２とａｄｄ２）が命令キャッシュ５２から読み出され，Ｆ１とＦ０に保持される。０命令発行で，サイクル３でも命令が実行されないとする。
【００７５】サイクル３でａｄｄ３とａｄｄ４が命令キャッシュ５２から読み出され，上位バイトのａｄｄ４はＴに保持され，下位バイトのａｄｄ３はＰ０に保持される。サイクル３で２命令発行がなされる。
【００７６】サイクル４でｌｏａｄ１とａｄｄ１がそれぞれＤ１，Ｄ０からスロット１とスロット０に出力される。そして，ｌｏａｄ２とａｄｄ２がそれぞれＤ１，Ｄ０に移動する。ａｄｄ３とａｄｄ４はそれぞれＦ０，Ｐ０に移動する。命令キャッシュ５２から，新たにｌｏａｄ３とｌｏａｄ５が読み込まれる。ｌｏａｄ３はａｄｄ４と境界が連続しているのでＰ１に書き込まれ，ｌｏａｄ４はＴに書き込まれる。ここで，２命令発行がなされる。
【００７７】サイクル５で，２命令発行に従って，ｌｏａｄ２とａｄｄ２がそれぞれＤ１，Ｄ０からスロット１，スロット０に出力される。そして，ａｄｄ３，ａｄ

ｄ４はそれぞれＤ０，Ｐ０に移動する。また，ｌｏａｄ３，ｌｏａｄ４はそれぞれＦ１，Ｐ１に移動する。ここで，次の命令ｌｏａｄ５とａｄｄ５を読み出す余地がないので，新たに命令は読み出されない。ここで１命令発行がなされる。

【００７８】サイクル６で，１命令発行に従って，ａｄｄ３がＤ０から出力され，ａｄｄ４がＤ０に移動する。そして，ｌｏａｄ３，ｌｏａｄ４がそれぞれＤ１，Ｆ１に移動する。命令キャッシュ５２から新たに，ｌｏａｄ５とａｄｄ５が読み出され，それぞれＰ１，Ｐ０に保持される。ここて２命令発行がなされる。

【００７９】サイクル７で，ｌｏａｄ３とａｄｄ４がそれぞれＤ０，Ｄ１からスロット１，スロット０に出力される。そして，ｌｏａｄ４がＤ１に移動し，ｌｏａｄ５，ａｄｄ５がそれぞれＦ１，Ｆ０に移動する。命令キャッシュ５２からｌｏａｄ６，ａｄｄ６が読み出され，それぞれＰ１，Ｐ０に保持される。ここで１命令が発行される。

【００８０】

【発明の効果】本発明によれば，マイクロプロセッサの処理速度に大きく影響する演算部Ａ，演算部Ｂのデコードステージでのスロット選択を行わない。そのため，デコードステージで処理が遅延されることがないので，能率的にパイプライン処理を行うことができ，マイクロプロセッサ全体の動作が高速化される。また，命令バッファの命令処理もフラグにより簡単に制御できる。

【図面の簡単な説明】

【図１】本発明の基本構成を示す図である。

【図２】本発明の実施例を示す図である。

【図３】本発明のフラグ生成のタイムチャートを示す図である。

【図４】本発明の命令バッファの実施例を示す図である。

【図５】本発明の命令キャッシュに保持されている命令の例を示す図である。

【図６】本発明の動作例１を示す図である。

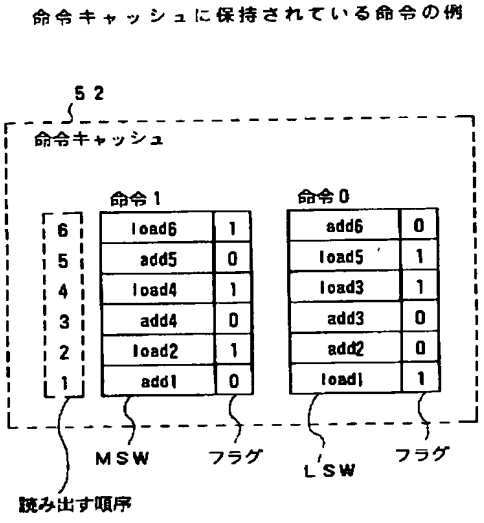
【図７】本発明の動作例２を示す図である。

【図８】従来のマイクロプロセッサの構成を示す図である。

【符号の説明】

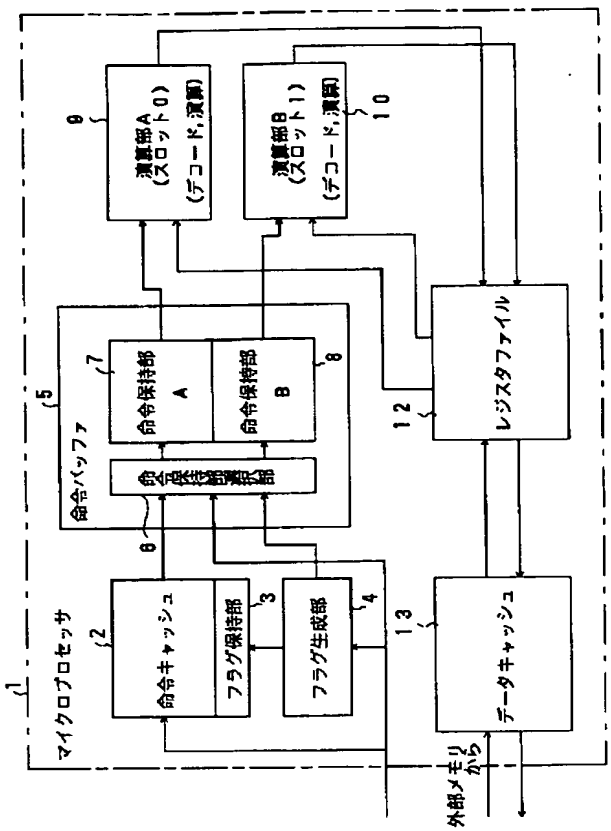
１：マイクロプロセッサ
２：命令キャッシュ
３：フラグ保持部
４：フラグ生成部
５：命令バッファ
６：命令保持部選択部
７：命令保持部Ａ
８：命令保持部Ｂ
９：演算部Ａ
１０：演算部Ｂ
１２：レジスタファイル
１３：データキャッシュ

【図５】



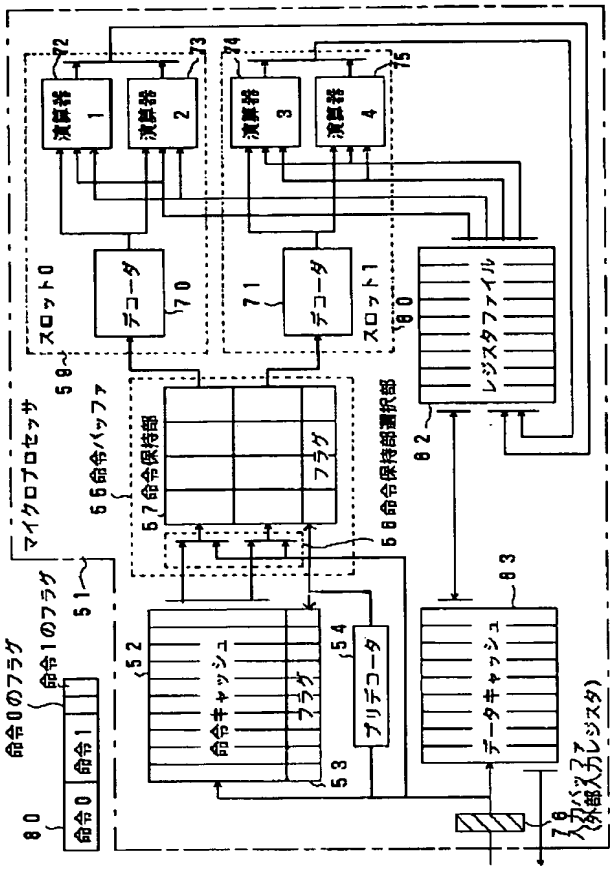
【図１】

本発明の基本構成



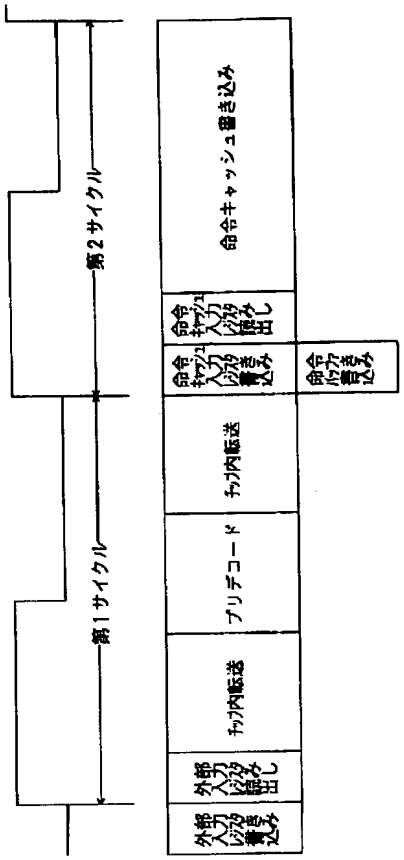
【図２】

本発明の実施例



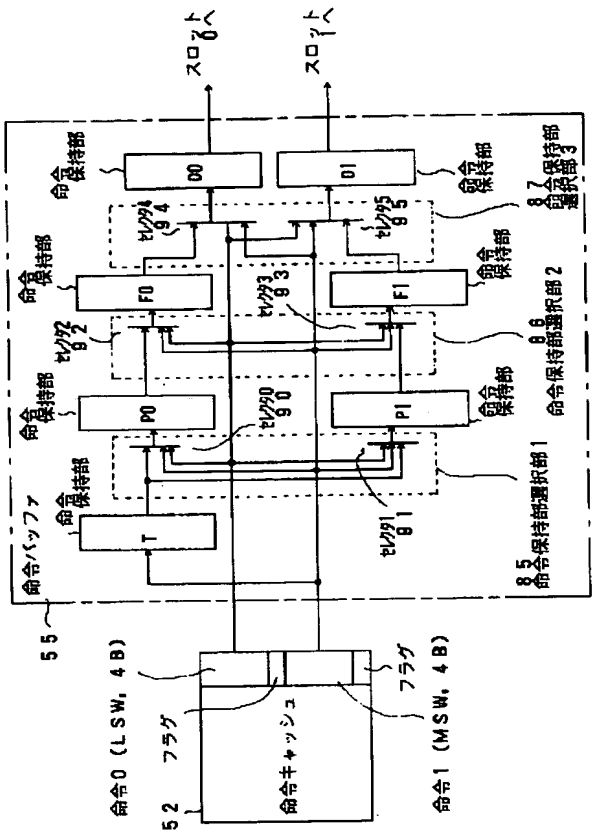
【図３】

本発明のフラグ生成のタイムチャート

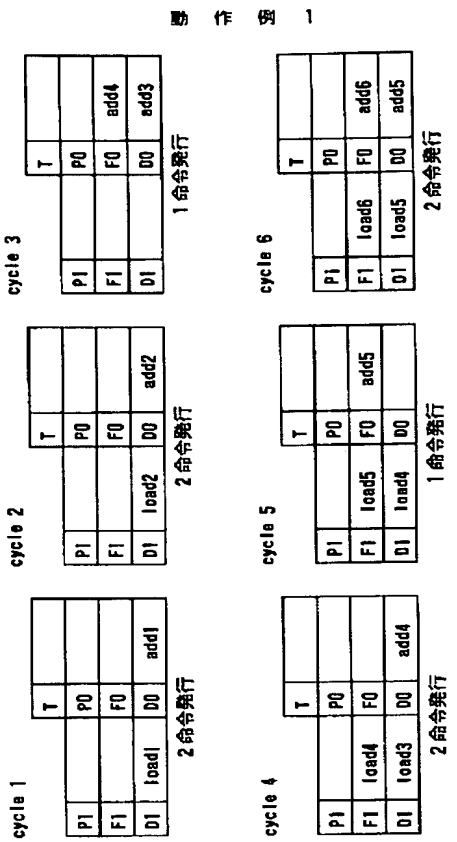


【図４】

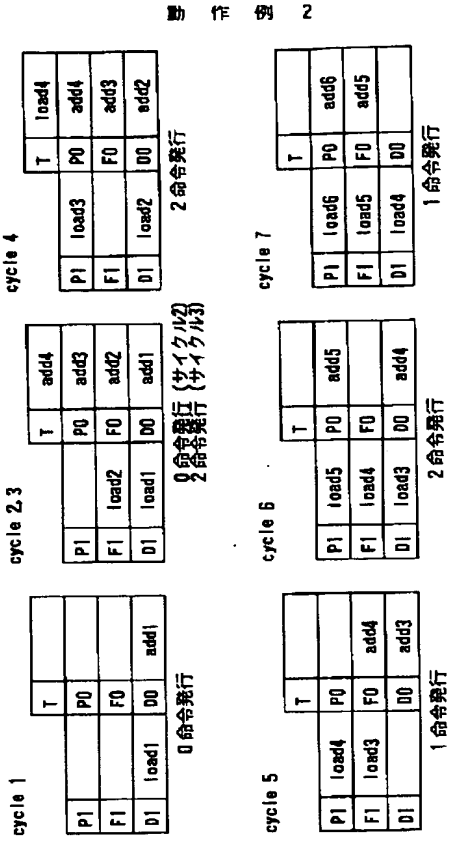
本発明の命令バッファの実施例



【図６】

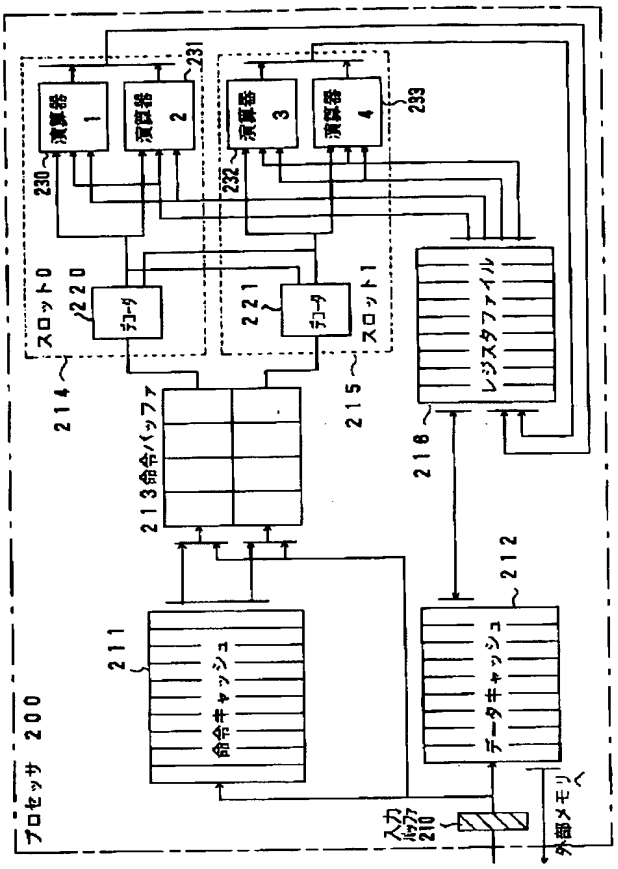


【図７】



【図８】

従来のマイクロプロセッサの構成



PATENT ABSTRACTS OF JAPAN

(11)Publication number : 08-044563

(43)Date of publication of application : 16.02.1996

(51)Int.Cl. G06F 9/38
G06F 9/38
G06F 12/08

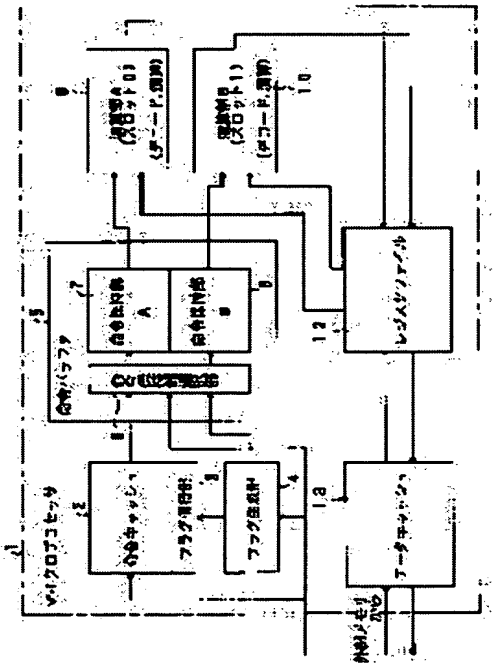
(21)Application number : 06-178129

(22)Date of filing : 29.07.1994

(71)Applicant : FUJITSU LTD

(72)Inventor : KUBOSAWA HAJIME

(54) MICROPROCESSOR



(57)Abstract:

PURPOSE: To increase the operation speed of a microprocessor by holding an instruction in an instruction holding part corresponding to the operation part, where this instruction is processed, not to select a slot in the stage of decoding.

CONSTITUTION: A flag generation part 4 decodes the instruction inputted from an external memory to discriminate the classification or the instruction. The instruction us read out from an instruction cache 2 into an instruction buffer 5 and is held there. The instruction, which is inputted from the external memory because being absent in the instruction cache 2, is inputted to the instruction buffer 5 after the flag is added by the flag generation part 4, and this instruction is held in the instruction cache 2. In this case, an instruction holding part selection part 6 discriminates the flag; and if the flag is '0', the instruction buffer 5 is selected, and the instruction is held in an instruction buffer A. If the flag is '1', the instruction holding part selection part 6 selects an instruction holding part B8, and the instruction is held in this part B8.